Supplementary Material for *DR-Tune: Improving Fine-tuning of Pretrained Visual Models by Distribution Regularization with Semantic Calibration*

Nan Zhou[1,2] Jiaxin Chen[2] Di Huang[1,2,3*]
[1]State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
[2]School of Computer Science and Engineering, Beihang University, Beijing, China
[3]Hangzhou Innovation Institute, Beihang University, Hangzhou, China
{zhounan0431,jiaxinchen,dhuang}@buaa.edu.cn

In this document, we describe more details about the datasets and the settings of hyper-parameters used for evaluation in Sec. A. Additionally, we summarize the overall pipeline of the proposed DR-Tune framework in Sec. B, and provide more analysis, semantic segmentation results as well as quantitative results in Sec. C, Sec. D and Sec. E, respectively. Finally, we discuss the limitations in Sec. F.

A. Details on Datasets and Hyper-parameters.

In Sec. 4 of the main body, we briefly summarize the datasets used for evaluation, including **ImageNet20** [4, 10], **CIFAR10 & 100** [13], **DTD** [3], **Caltech101** [5], Stanford **Cars** [12], Oxford-IIIT **Pets** [21], Oxford 102 **Flowers** [19], FGVC **Aircraft** [17], **SVHN** [18] and **Sun397** [24]. As a supplement, we describe more details in this section.

ImageNet20 is a subset of the large-scale ImageNet dataset [4], which contains 26,348 images from 20 categories. It is collected by combining an easy-to-classify dataset Imagenette and a hard-to-classify dataset Imagewoof [10]. On this dataset, 18,494 images are used for training and the rest 7,854 images are utilized for evaluation.

CIFAR10 & 100 [13] are two widely used datasets containing natural objects from 10 and 100 categories, respectively. They are both divided into a subset of 50,000 images for training and a subset of 10,000 images for evaluation.

Describable Textures Dataset (DTD) [3] is a texture dataset, consisting of 5,640 images organized according to a list of 47 categories inspired from human perception. 3,760 images are used for training and the remaining 1,880 images are adopted for evaluation.

The **Caltech101** dataset [5] includes 9,146 images from 101 distinct categories, each of which contains 40 to 800 images. We use 3,060 images and 6,084 images for training and evaluation, respectively.

Stanford **Cars** [12] is a fine-grained dataset, which contains 16,185 images of 196 different types of cars. This dataset is split into a set of 8,144 images for training and a set of 8,041 images for evaluation.

Algorithm 1: The overall pipeline of DR-Tune.

```
Input: The pretrained encoder $f_{\boldsymbol{\theta}^p}$, the size of the
       memory bank $K$ and the batch size $B$.
Output: The fine-tuned downstream encoder $f_{\boldsymbol{\theta}^d}$
        and the classification head $g_{\boldsymbol{\phi}^d}$.
Initialization: Set $\boldsymbol{\theta}^d := \boldsymbol{\theta}^p$, randomly initialize
  $\boldsymbol{\phi}^d$, and fill the memory banks $\boldsymbol{\mathcal{M}}^p$ and $\boldsymbol{\mathcal{M}}^d$ with
  the pretrained features.
while not converge do
    Sample a mini-batch $\{\boldsymbol{x}_i^d, y_i\}_{i=1}^B$.
    for $i \in \{1, \cdots, B\}$ do
        Extract the pretrained and downstream
          features for $\boldsymbol{x}_i^d$ as follows:
          $\boldsymbol{z}_i^p = f_{\boldsymbol{\theta}^p}(\boldsymbol{x}_i^d), \boldsymbol{z}_i^d = f_{\boldsymbol{\theta}^d}(\boldsymbol{x}_i^d)$.
    end
    Calculate the rotation matrix $\boldsymbol{R}$ via SVD [22].
    Compute the class-level translations as below:
    for $c = 1$ to $C$ do
        Calculate $\boldsymbol{\mu}_c^p$ based on $\boldsymbol{\mathcal{M}}^p$ by Eq. (8).
        Calculate $\boldsymbol{\mu}_c^d$ based on $\boldsymbol{\mathcal{M}}^d$ by Eqs. (9)-(10).
        Compute the $c$-th translation vector as below
        $\boldsymbol{\delta}_c = \boldsymbol{\mu}_c^d - \boldsymbol{\mu}_c^p$.
    end
    Calibrate the memory bank $\boldsymbol{\mathcal{M}}^p$ via Eq. (12).
    Update $\boldsymbol{\theta}^d$ and $\boldsymbol{\phi}^d$ by optimizing Eq. (14).
    Update $\boldsymbol{\mathcal{M}}^p$/$\boldsymbol{\mathcal{M}}^d$ by $\boldsymbol{z}_i^p$/$\boldsymbol{z}_i^d$, respectively.
end
```

Oxford-IIIT **Pets** [21] consists of the images captured from 37 kinds of pets, of which each class roughly includes 200 images. This dataset exhibits large variations in scale, pose and lighting. We use 3,680 images for training and the rest 3,369 images for evaluation.

[*] Corresponding author.

Hyper-parameter	ImageNet20	CIFAR10	CIFAR100	DTD	Caltech101	Cars	Pets	Flowers	Aircraft
Epochs	100							200	100
lr schedule	linear decay	cosine decay							
lr for the encoder	0.01	0.01	0.01	0.01	0.1	0.1	0.01	0.01	0.1
lr for the head	0.33	0.33	0.33	0.33	0.1	0.1	0.17	0.13	0.1
The size K of memory banks	2048	2048	2048	2048	2048	2304	1024	768	2048
The batch size B	64								
Weight decay factor	10^{-4}								
Momentum factor	0.9								

Table A: Details about the hyper-parameters used for comparison with the fine-tuning methods based on the *self-supervised pretrained model*, corresponding to Table 1 of the main body. 'lr' is the abbreviation of 'learning rate'.

Hyper-parameter	CIFAR100†	Caltech101†	DTD†	Flowers†	Pets†	SVHN	Sun397
Epochs	100	300	100				
lr schedule	cosine decay						
lr for the encoder	0.01	0.01	0.01	0.01	0.01	0.01	0.01
lr for the head	0.17	0.02	0.1	0.33	0.1	0.1	0.1
The size K of memory banks	512	128	32	2048	256	128	2048
The batch size B	32						
Weight decay factor	10^{-4}	10^{-3}	10^{-3}	10^{-4}	10^{-4}	10^{-3}	10^{-4}
Momentum factor	0.9						

Table B: Details about the hyper-parameters used for comparison with the fine-tuning methods based on the *supervised pretrained model*, corresponding to Table 2 of the main body. 'lr' is the abbreviation of 'learning rate'. '†' refers to the training/test split setting as in [26].

Oxford 102 **Flowers** [19] contains 7,370 flower images from 102 different categories. 6,552 images are used for training and 818 images for evaluation.

The FGVC **Aircraft** [17] is a fine-grained dataset, which contains 10,000 images from 100 different types of aircraft models. We split this dataset into a subset of 6,667 images for training and the remaining 3,333 images for evaluation.

SVHN is obtained from house numbers in Google Street View images, including 73,257 training images and 26,032 test images of size 32x32 from 10 classes. By following the training/test split setting as in [26], we adopt 1,000 images for training and 26,032 images for evaluation.

Sun397 [24] is a scene understanding benchmark with 76,128 training images and 21,750 test images of 397 categories. Following the training/test split setting as in [26], we adopt 1,000 images for training and 21,750 images for evaluation.

Settings of hyper-parameters. As depicted in Sec. 4.3 of the main body, we compare DR-Tune with the state-of-the-art under two different settings, *i.e.* the one based on the self-supervised pretrained model and the other based on the supervised pretrained model. The corresponding settings of hyper-parameters are summarized in Table. A and Table. B, respectively.

B. Overall Pipeline of DR-Tune

In Sec. 3 of the main body, we elaborate the technical details on the main components of Dr-Tune. We additionally summarize the overall pipeline of DR-Tune in Algorithm 1.

C. More Analysis on DR-Tune

In this section, we conduct a more detailed study on how DR-Tune contributes to the performance gain by analyzing the encoder as well as the classification head on the CIFAR-10 benchmark. We also analyze some detailed designs in the SC module and compare DR-Tune with knowledge distillation (KD). Furthermore, we report the runtime cost and standard errors.

On the classification head. In this case, we take a counterpart, which is composed of a frozen downstream encoder fine-tuned by CE-tuning and a classification head randomly initialized. As shown in Fig. A (a) and (b), the classification head is trained by the standard Cross-Entropy loss (*i.e.* $\mathcal{L}_{\text{CE}}$) and the one used in DR-Tune (*i.e.* $\mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{R}_{\text{DR}}$), respectively; and we can observe that the top-1 accuracy is improved from 96.52% to 96.72%, indicating that $\mathcal{R}_{\text{DR}}$ leads to a better classification head.

On the encoder. We compare two models that are depicted in Fig. A (a) and (c), both of which have a frozen

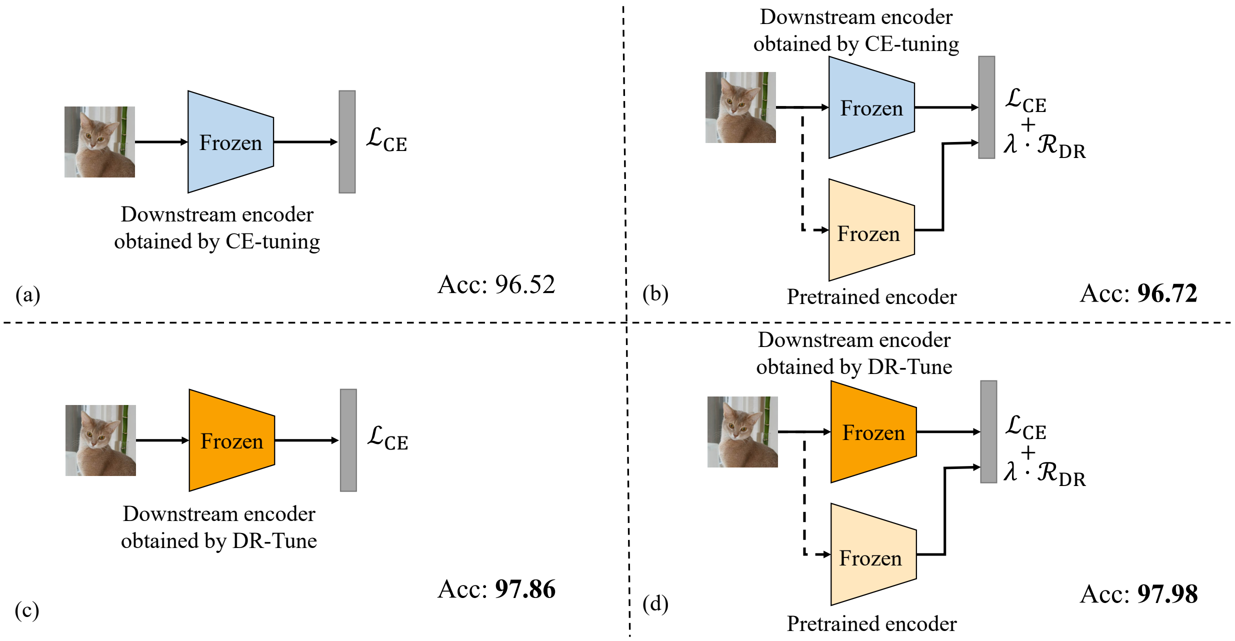

Frozen
Downstream encoder obtained by CE-tuning
$\mathcal{L}_{\text{CE}}$
(a)
Acc: 96.52
Downstream encoder obtained by CE-tuning
Frozen
Frozen
Pretrained encoder
$\mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{R}_{\text{DR}}$
(b)
Acc: **96.72**
Frozen
$\mathcal{L}_{\text{CE}}$
Downstream encoder obtained by DR-Tune
(c)
Acc: **97.86**
Downstream encoder obtained by DR-Tune
Frozen
Frozen
Pretrained encoder
$\mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{R}_{\text{DR}}$
(d)
Acc: **97.98**

Figure A: Illustration of different learning strategies: (a) The baseline CE-Tuning; (b) Training the classification head by optimizing $\mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{R}_{\text{DR}}$; (c) Applying the downstream encoder generated by DR-Tune; (d) Combining the settings in (b) and (c).

Operation	Imagenet20	CIFAR10	Pets
CLR	95.82	97.75	90.19
SA	95.77	97.79	90.24
GR (w/o SA)	95.85	97.82	89.56
GR (**Ours**)	**96.03**	**98.03**	**90.57**

Table C: Top-1 accuracies (%) of different operations in the SC module.

Method	Reference	Teacher	Caltech101	DTD
CE-tuning	-	-	93.38	68.62
KD [23]	NeurIPS'14	ResNet-50†	94.46	72.66
		ResNet-101†	93.68	74.42
		ResNet-101*	95.04	76.86
RKD [20]	CVPR'19	ResNet-50†	93.66	69.10
MLD [11]	CVPR'23	ResNet-50†	94.90	72.82
DR-Tune	**Ours**	-	**95.10**	**77.97**

Table D: Top-1 accuracies (%) of KD and DR-Tune with ResNet-50 as student network. †: pretrained by InfoMin; *: supervised pretraining.

downstream encoder and a randomly initialized classification head and are trained by $\mathcal{L}_{\text{CE}}$. Their difference lies in that the downstream encoder is fine-tuned by CE-tuning or by DR-Tune, and this change improves the top-1 accuracy from 96.52% to 97.86%, showing that DR-Tune facilitates the training of a stronger encoder.

As shown in Fig. A (d), when we combine the settings in Fig. A (b) and (c), the improved encoder and classification head finally reach the top-1 accuracy of 97.98%, highlighting the effectiveness of DR-Tune.

On the SC module. Global rotation (GR) is performed in the SC module to alleviate the semantic drift. We explore some different designs for this. (1) Rotation is performed around the category center of each class, *i.e.* class-level rotation (CLR). (2) Replace the rotation operation by aligning the L2-norm between pretrained and downstream features, *i.e.* scale alignment (SA). As shown in Table C, CLR does not lead to a gain, but takes $C-1$ times more operations than GR (C: number of classes). We thus adopt GR in implementation. The performance of SA is not as good as GR in most cases, but using SA with GR can boost the performance, indicating that using both rotation and scale alignment is a better option.

Comparison to knowledge distillation. The Knowledge distillation (KD) based methods utilize a frozen pretrained teacher network to guide the student network, which has a similar framework with DR-Tune. We thus compare DR-Tune to some representative KD-based methods: 1) logit distillation including KD [9] and MLD [11] and 2) feature distillation *i.e.* RKD [20]. Despite sharing the

Method	Train		Test		
	Latency↓ (ms)	Memory↓ (GB)	Latency↓ (ms)	Memory↓ (GB)	Accuracy↑ (%)
CE-tuning	73.55	7.64	66.68	4.22	87.76
Core-tuning [64]	151.92	22.22	67.04	4.22	90.47
DR-Tune (**Ours**)	167.50	8.41	66.49	4.22	91.35

Table E: Comparison of runtime cost and accuracy.

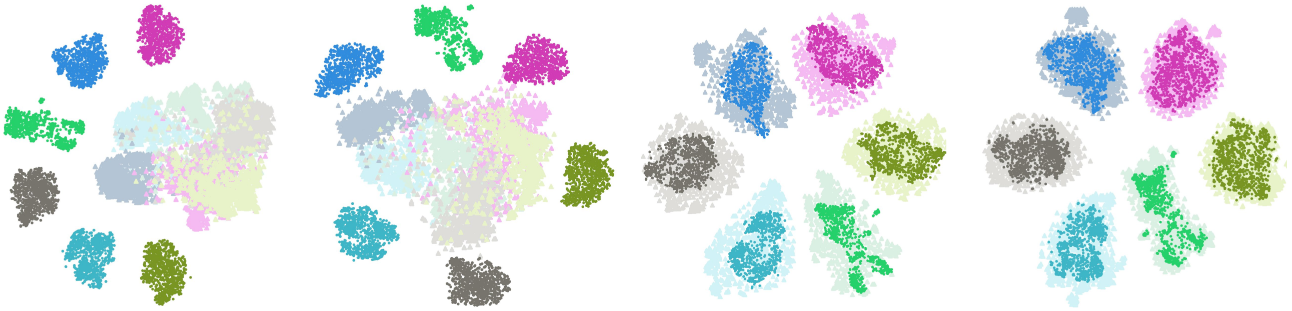

(a) w/o. Semantic Calibration (b) Global Rotation (c) Class-Level Translation (d) w. Semantic Calibration

Figure B: *t*-SNE [23] visualization of the pretrained and downstream features on CIFAR10 from the first 6 classes. Different colors indicate different classes, and points with low/high brightness denote the pretrained/downstream features, respectively.

same spirit of using pretrained models as regularizers, the KD-based methods ignore the semantic drift issue and impose constraints on the whole downstream model instead of the task head, which may degrade the performance. As an empirical study, Table D shows that all the KD-based methods boost the accuracy of the baseline CE-tuning, but perform worse than DR-Tune when using the same teacher ResNet-50 pretrained by InfoMin. We then evaluate KD using different teachers with various backbones and pretraining schemes. As displayed, larger teacher models deliver further improvements to KD, but the results are still not as good as those of DR-Tune.

On the runtime cost. We report the latency and memory for CE-tuning, Core-tuning and DR-Tune, evaluated using the same NVIDIA V100 GPU with a batch size of 64, based on ResNet-50 pretrained by MoCo-v2. As in Table E, DR-Tune has relatively higher training latency compared to CE-tuning, due to extra computation in DR and SC. Core-tuning suffers much more memory usage, as it employs extra parameters and the feature mixture strategy. However, DR-Tune takes a similar cost to CE-tuning in testing, since DR and SC are not used in this phase. Besides, DR-tune delivers remarkably higher accuracies, thus reaching a better balance between efficiency and accuracy for deployment.

On the standard errors. In Table 1 and Table 2 of the main body, we report the mean results after repeating the experiments for three times with different random seeds on each dataset, omitting the standard errors for succinctness. In this supplement, we provide the standard errors to validate the robustness. Note that the counterparts including Linear probing, Adapter, Bias, VPT and SSF in Table 2 do NOT report the standard errors. Therefore, we only report the standard errors of DR-Tune and the re-implemented baseline Core-tuning. The results are summarized in Table F and Table G, showing that our method steadily reaches moderately small standard errors on different datasets and settings.

D. Results on Semantic Segmentation

In this section, we evaluate the generalizability of DR-Tune on the semantic segmentation task beyond classification.

Following the same setting as [28] does, we evaluate DR-Tune on semantic segmentation. Since only CE-tuning and Core-tuning report the results on this task among the counterparts in Table 1, we take them for comparison. As Table H displays, DR-Tune clearly outperforms them, showing its generalizability beyond classification.

Method	ImageNet20	CIFAR10	CIFAR100	DTD	Caltech101	Cars	Pets	Flowers	Aircraft	Avg.
CE-tuning	88.28±0.47	94.70±0.39	80.27±0.60	71.68±0.53	91.87±0.18	88.61±0.43	89.05±0.01	98.49±0.06	86.87±0.18	87.76
L2SP [25]	88.49±0.40	95.14±0.22	81.43±0.22	72.18±0.61	91.98±0.07	89.00±0.23	89.43±0.27	98.66±0.20	86.55±0.30	88.10
DELTA [15]	88.35±0.41	94.76±0.05	80.39±0.41	72.23±0.23	92.19±0.45	88.73±0.05	89.54±0.48	98.65±0.17	87.05±0.37	87.99
M&M [27]	88.53±0.21	95.02±0.07	80.58±0.19	72.43±0.43	92.91±0.08	88.90±0.70	89.60±0.09	98.57±0.15	87.45±0.28	88.22
BSS [2]	88.34±0.62	94.84±0.21	80.40±0.30	72.22±0.17	91.95±0.12	88.50±0.02	89.50±0.42	98.57±0.15	87.18±0.71	87.94
RIFLE [14]	89.06±0.28	94.71±0.13	80.36±0.07	72.45±0.30	91.94±0.23	89.72±0.11	90.05±0.26	98.70±0.06	87.60±0.50	88.29
SCL [7]	89.29±0.07	95.33±0.09	81.49±0.27	72.73±0.31	92.84±0.03	89.37±0.13	89.71±0.20	98.65±0.10	87.44±0.31	88.54
Bi-tuning [29]	89.06±0.08	95.12±0.15	81.42±0.01	73.53±0.37	92.83±0.06	89.41±0.28	89.90±0.06	98.57±0.10	87.39±0.01	88.58
Core-tuning [28]	92.73±0.17	97.31±0.10	84.13±0.27	75.37±0.37	93.46±0.06	90.17±0.03	**92.36**±0.14	99.18±0.15	89.48±0.17	90.47
SSF* [16]	94.72±0.07	95.87±0.10	79.57±0.02	75.39±0.66	90.40±0.17	62.22±0.21	84.89±0.17	92.15±0.55	62.38±0.55	81.95
DR-Tune (Ours)	**96.03**±0.11	**98.03**±0.04	**85.47**±0.08	**76.65**±0.07	**95.77**±0.12	**90.60**±0.15	90.57±0.09	**99.27**±0.10	**89.80**±0.09	**91.35**

Table F: Comparison of the top-1 accuracies (%) as well as the standard errors by using various fine-tuning methods based on the *self-supervised pretrained model*, *i.e.* ResNet-50 pretrained by MoCo-v2 on ImageNet. '*' indicates that the method is re-implemented. The best results are in **bold**.

Method	CIFAR100†	Caltech101†	DTD†	Flowers†	Pets†	SVHN	Sun397	Avg.
Core-tuning [28]	66.3±0.55	89.7±0.07	70.9±0.03	99.0±0.05	92.3±0.16	76.4±0.08	52.5±0.85	78.16
DR-Tune (Ours)	**81.1**±0.34	**92.8**±0.19	71.4±0.41	99.3±0.02	**92.4**±0.21	**92.0**±0.10	**54.5**±0.03	**83.36**

Table G: Comparison of the top-1 accuracies (%) as well as the standard errors by using various fine-tuning methods based on the *supervised pretrained model*, *i.e.* ViT-B pretrained on ImageNet. '*' indicates that the method is re-implemented. '†' refers to the training/test split setting as in [26]. The best results are in **bold**.

Method	MPA ↑	FWIoU ↑	MIoU ↑
CE-tuning	87.31	90.26	78.42
Core-tuning [64]	88.76	90.75	79.62
DR-Tune (**Ours**)	**89.90**	**90.81**	**79.93**

Table H: Results (%) on PASCAL VOC for semantic segmentation, using DeepLab-V3 [1] with ResNet-50 pretrained by MoCo-v2.

E. Qualitative Results

Visualization of the SC process. We provide visualization results on CIFAR10 to demonstrate the effectiveness of the transformations used in the SC module. As displayed in Fig. B, the pretrained feature distribution (low brightness) and the downstream counterpart (high brightness) clearly exhibit a semantic drift. Global rotation mitigates the misalignment of the overall shape as well as the overall center. Class-level translations align the centers for each class, further alleviating the semantic drift. We also add quantitative evaluations by adopting the Maximum Mean Discrepancy (MMD) [6] metric in Table I, showing that the distribution distance remarkably decreases.

Method	w/o. SC	w. SC (**Ours**)
$\mathrm{MMD}(\mathcal{Z}^p, \mathcal{Z}^d)$	1.478	0.028

Table I: Comparison in terms of MMD on CIFAR10.

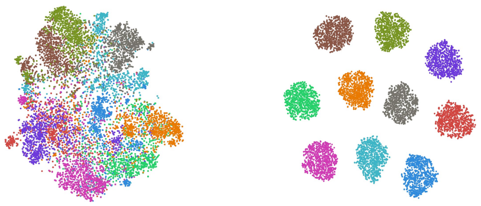


Figure C: t-SNE visualization of distributions of the pretrained (left) and downstream (right) features on CIFAR10.

Visualization of the feature distribution. In Sec 3.4 of the main body, due to the lack of supervision in the downstream task, the inter-class distribution of the pretrained feature is less discriminative than the downstream one. To make it more convincing, we visualize the distributions of the pretrained and downstream features on CIFAR10 in Fig. C, where the downstream ones are more discriminative.

Visualization of the training process. In Fig. D, we use t-SNE [23] to visualize the features of the training and testing sets from CIFAR10 [13] during training. We also use the S_Dbw score [8] to evaluate the inter-class density and intra-class variance of the learned features where a lower S_Dbw score is better. DR-Tune utilizes the prior knowledge that accelerates the convergence, and therefore a faster convergence process is observed compared to vanilla fine-tuning (*i.e.* CE-tuning), which only uses the pre-trained model for initialization. Besides, after training, the features obtained by DR-Tune have a lower S_Dbw score, indicating a more compact intra-class distribution and a more dispersed inter-class distribution.

F. Limitations

As discussed in Sec. C, DR-Tune suffers from a high training latency, due to computation of rotations by SVD in SC, which can be further improved by more efficient solutions. Besides, SC aligns the downstream and pretrained features by a global feature after average pooling for classification, ignoring spatial misalignment, which is crucial to spatio-sensitive tasks, *e.g.* object detection ans semantic segmentation, leaving room for gains.

References

[1] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 5

[2] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: batch spectral shrinkage for safe transfer learning. In *NeurIPS*, 2019. 5

[3] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 1

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[5] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR Workshop*, 2004. 1

[6] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *JMLR*, 2012. 5

[7] Beliz Gunel, Jingfei Du, Alexis Conneau, and Veselin Stoyanov. Supervised contrastive learning for pre-trained language model fine-tuning. In *ICLR*, 2020. 5

[8] Maria Halkidi and Michalis Vazirgiannis. Clustering validity assessment: Finding the optimal partitioning of a data set. In *ICDM*, 2001. 6

[9] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[10] Jeremy Howard. The imagenette and imagewoof datasets. https://github.com/fastai/imagenette, 2019. 1

[11] Y. Jin. Multi-level logit distillation. In *CVPR*, 2023. 3

[12] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013. 1

[13] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. *Tech Report, University of Toronto*, 2009. 1, 6

[14] Xingjian Li, Haoyi Xiong, Haozhe An, Cheng-Zhong Xu, and Dejing Dou. Rifle: backpropagation in depth for deep transfer learning through re-initializing the fully-connected layer. In *ICML*, 2020. 5

[15] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, and Jun Huan. Delta: deep learning transfer using feature map with attention for convolutional networks. In *ICLR*, 2018. 5

[16] Dongze Lian, Zhou Daquan, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. In *NeurIPS*, 2022. 5

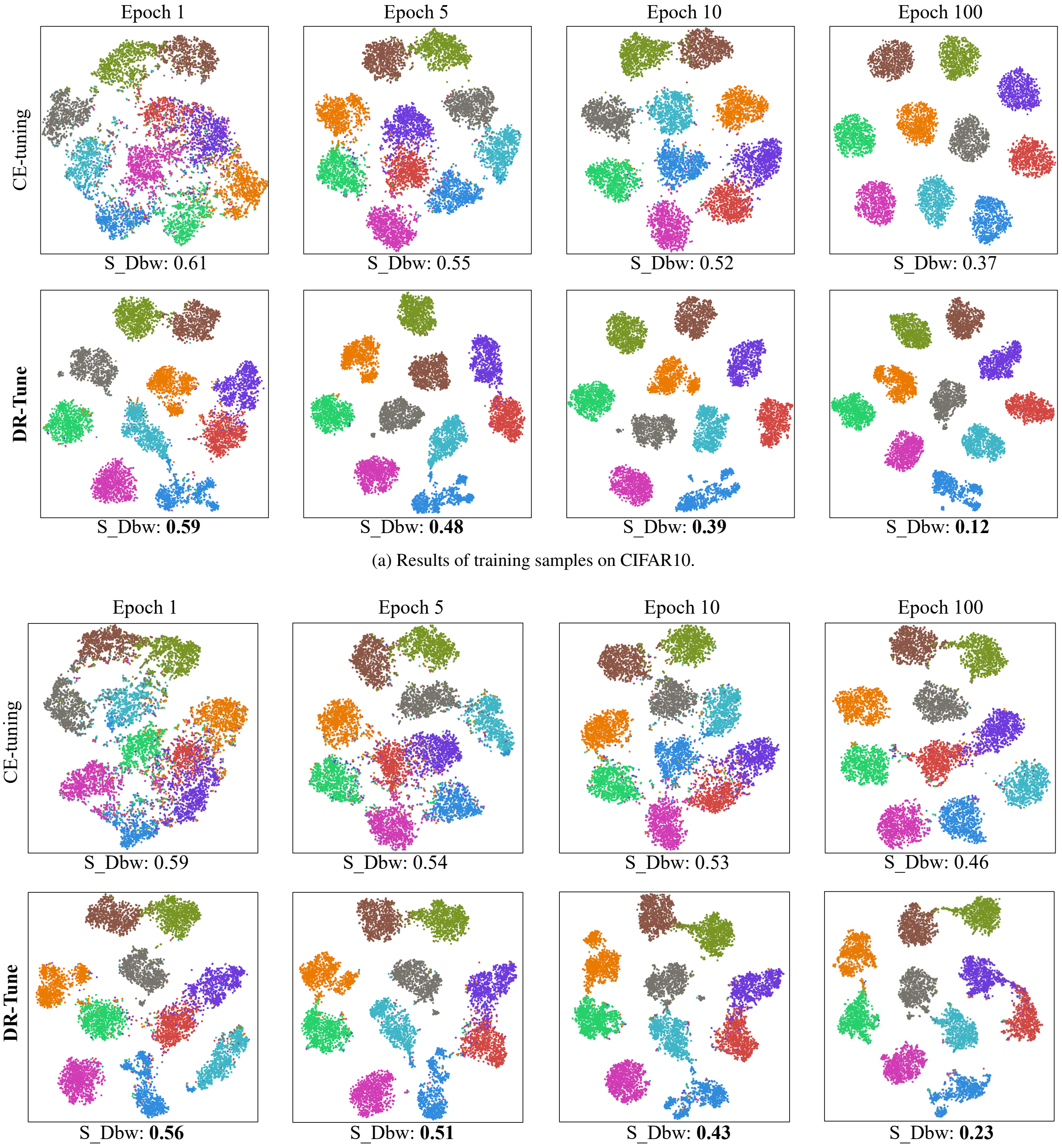
[17] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1, 2

[18] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 1

[19] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 1, 2

[20] W. Park. Relational knowledge distillation. In *CVPR*, 2019. 3

[21] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012. 1



(a) Results of training samples on CIFAR10.

(b) Results of testing samples on CIFAR10.

Figure D: *t*-SNE visualizattion and S_Dbw scores of the learned features on the CIFAR10 dataset: (a) on the training samples and (b) on the testing samples. CE-tuning refers to vanilla fine-tuning.

[22] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *Computing*, 1:1–5, 2017. 1

[23] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9:2579–2605, 2008. 4, 6

[24] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 1, 2

[25] Li Xuhong, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *ICML*, 2018. 5

[26] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov,

Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 2, 5

[27] Xiaohang Zhan, Ziwei Liu, Ping Luo, Xiaoou Tang, and Chen Loy. Mix-and-match tuning for self-supervised semantic segmentation. In *AAAI*, 2018. 5

[28] Yifan Zhang, Bryan Hooi, Dapeng Hu, Jian Liang, and Jiashi Feng. Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning. In *NeurIPS*, 2021. 4, 5

[29] Jincheng Zhong, Ximei Wang, Zhi Kou, Jianmin Wang, and Mingsheng Long. Bi-tuning of pre-trained representations. *arXiv preprint arXiv:2011.06182*, 2020. 5